Brief Overview of Future Technologies and Evaluation Efforts at ORNL

02 October 2006

Our Team and Collaborators

- Future Technologies Group
  - Sadaf Alam
  - Richard Barrett
  - Nikhil Bhatia
  - Jeff Kuehn
  - Collin McCurdy
  - Jeremy Meredith
  - Ken Roche
  - Philip Roth
  - Olaf Storaasli
  - Jeffrey Vetter
  - Weikuan Yu
- Collaborators
  - Jacob Barhen
  - Pat Worley
  - Pratul Agarwal
  - Hong Ong
  - Core universities (UT, GT, Duke, NCSU, …)
  - SciDAC PERC Team
  - DARPA HPCS Team
  - DoD HPCMP
  - Georgia Tech CSE Dept, CERCS
  - Vendors
  - Many others…
- Support Scientific Computation through:
  - Performance Analysis
  - System Evaluation
  - Modeling
  - HEC Algorithm/Software R&D
- Experimental Computing Lab (ExCL)
  - Examine new/emerging technologies
  - FPGAs
  - Optical processors
  - GPUs
  - Array processors
  - Multicore processors
  - …
- System software: OS, performance tools, runtime systems

Motivation

- ⇨ ITRS predicts that Moore's Law will be coming to an end in 2012-2015
  - – Recent performance gains at chip level are principally derived from Moore's Law
- ⇨ Low sustained performance on important applications causing HEC community to reconsider system designs
  - – System balance not tuned for specific HEC applications
- ⇨ Infrastructure (power, cooling, space) impacting system design
  - – Tempers performance gains of conventional microprocessors
- ⇨ Market trends drive vendors to favor issues other than performance (laptop battery life, etc)
  - – Culture of "Good-enough computing" -- *Economist*

Intel's CTO: "Pentium PC May Need the Power of a Nuclear Reactor"

Computer Chips to be Hotter than the Sun?

by Anton Shilov
02/20/2004 | 06:27 PM

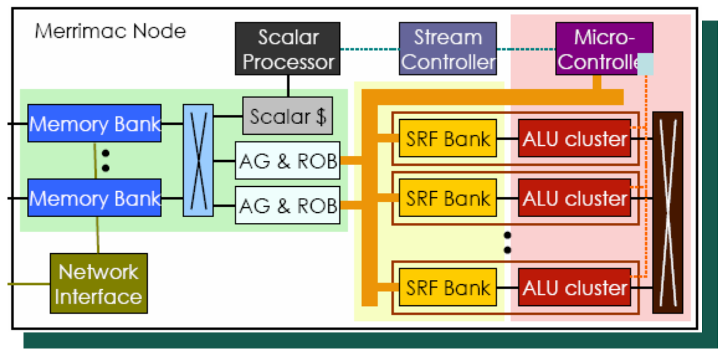
Motivation (cont)

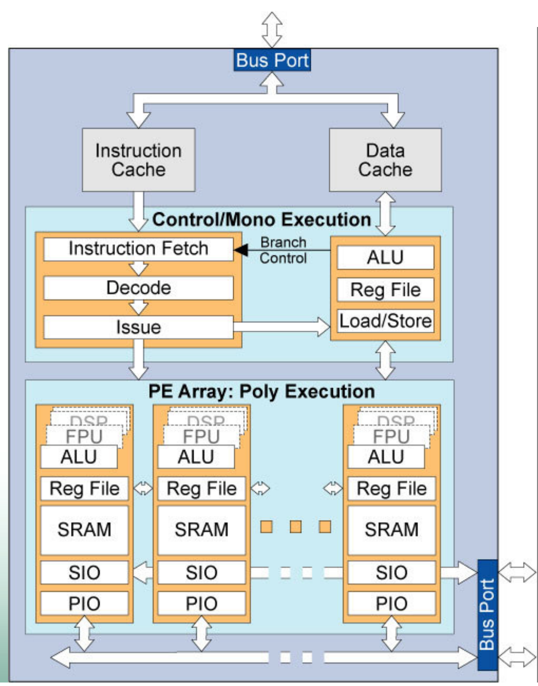
- ⇨ New constraints define a new utility function
  - Sustained performance, reliability, power, floorspace
- ⇨ Current architectural trends
  - Major vendors moving toward multicore architectures
- ⇨ Commodity constraints on HEC driving a rebirth of computer architecture

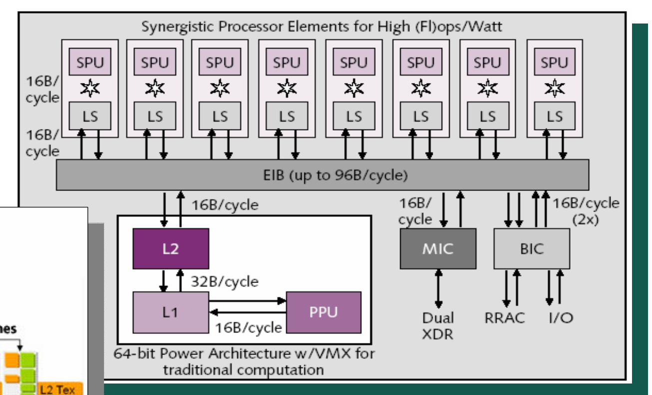
Alternative Architectures Offer Different Design Points

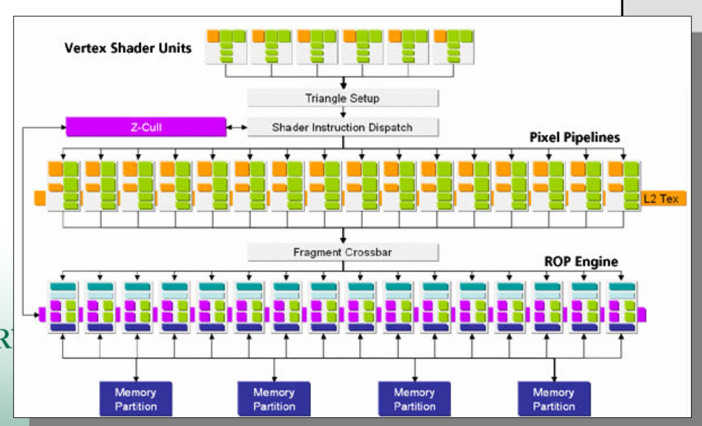
- GPUs
- FPGAs
- Mulithreaded processors
- Game processors
- Streaming processors

- Physics and AI chips





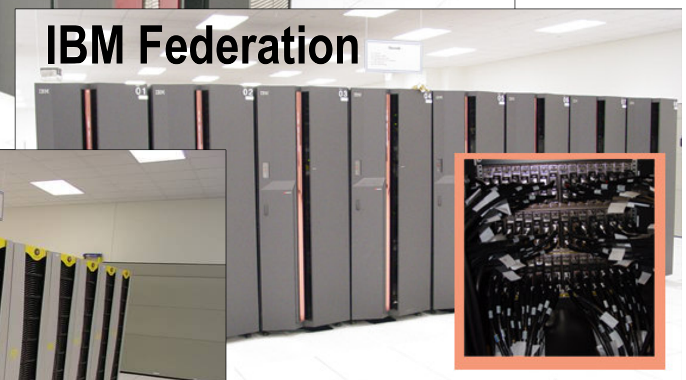


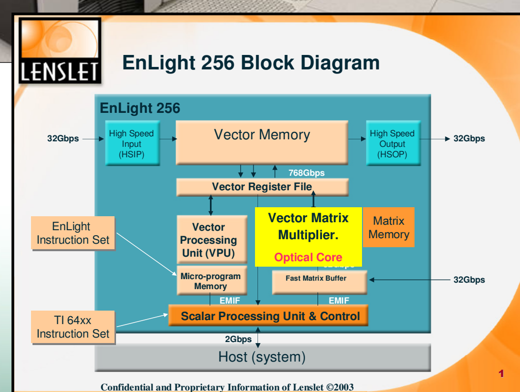


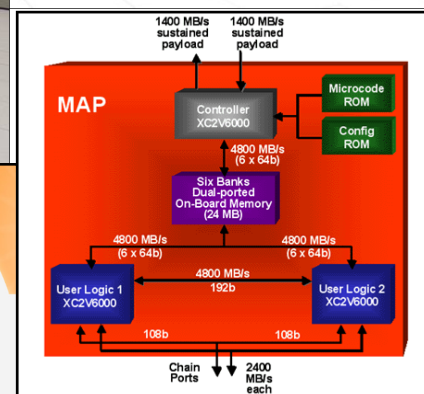
UT-BATTELLE

Investigating Diverse Architectures

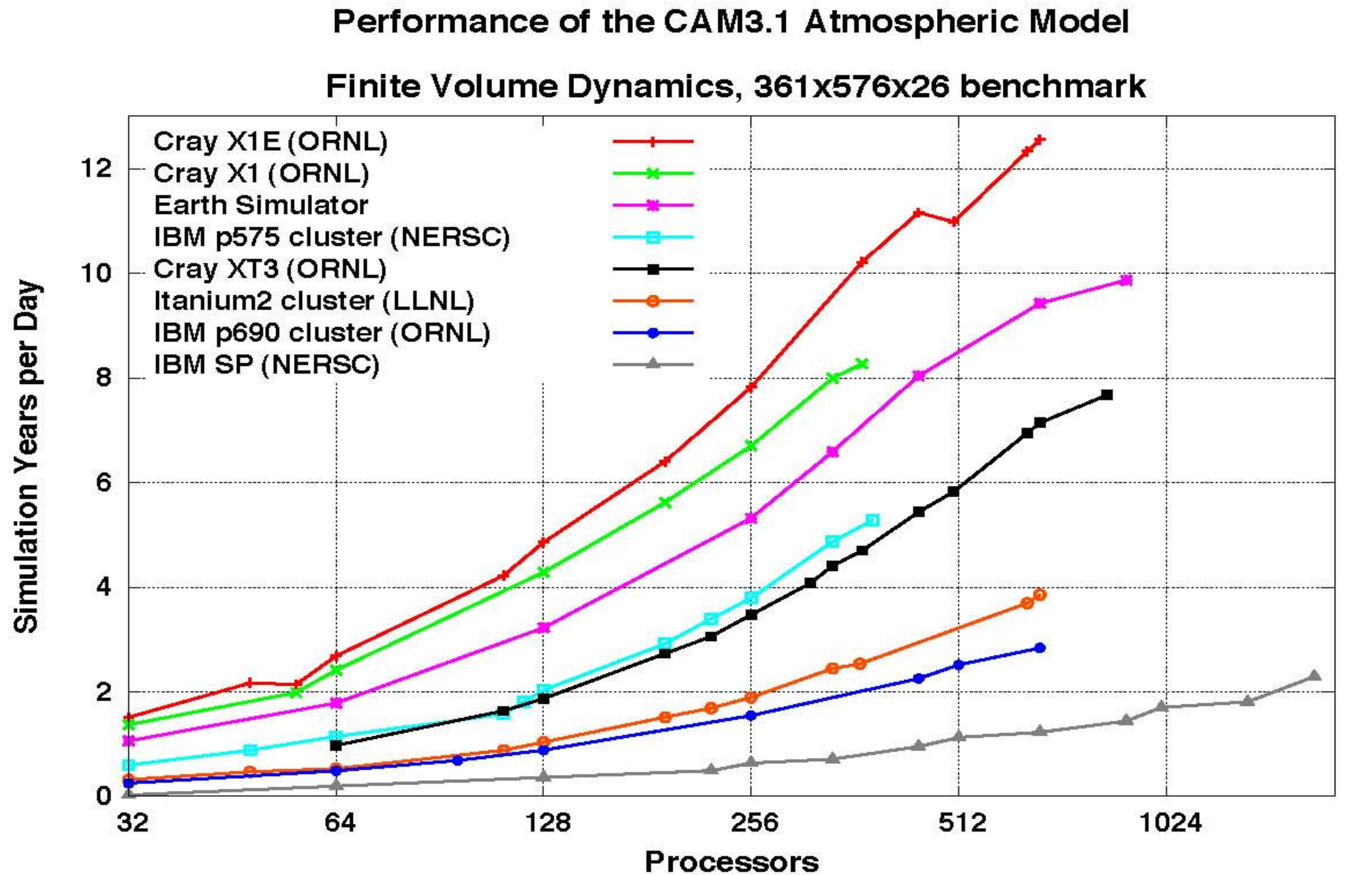
- ⇨ Cray X1e, XD1, XT3 (Red Storm)
- ⇨ IBM BlueGene/L
- ⇨ IBM SP3, p655
- ⇨ SGI Altix
- ⇨ Intel Itanium, Xeon
- ⇨ IBM POWER5
- ⇨ IBM Cell
- ⇨ FPGAs
- ⇨ GPUs
- ⇨ Optical processors
- ⇨ Multithreading
- ⇨ Array processors, etc.
- ⇨ Processors-in-memory
- ⇨ This diversity makes the problem much more interesting!







Evaluations are Application Grounded

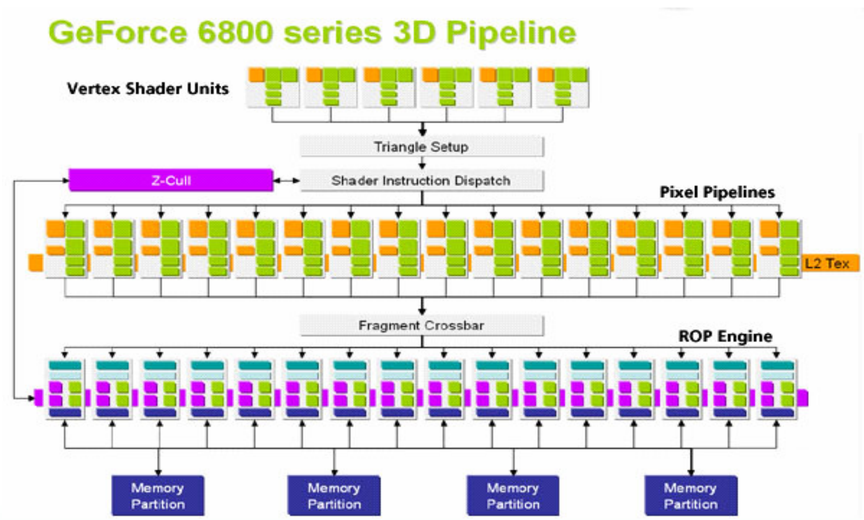


Data Courtesy P.H.Worley

GPUs – current technology with a twist

- What are they?
  - Graphics Processing Units
  - Most "video cards" contain GPU, RAM
    - Usually AGP / PCIe cards
    - Up to 512 MB RAM, though most contain less
  - Accelerate common 3D graphics operations
    - Vertex transformations
    - Polygon rasterization

- Architecture
  - Vertex and Pixel (Fragment) stages
    - implicit parallelism
    - differing programmability
  - Poor support for scatter ops
  - High internal bandwidth
  - Example: NV 6800 GT
    - 6 vertex pipelines
    - 16 fragment pipelines



GPUs – Motivation and Outlook

- Motivation
- High end consumer cards:
  - Very high speeds
  - Low price (~$500)
  - Improving faster than CPUs
  - Mass market driven (gaming)
- Example:
  - 60 GFLOPS, 16 pipes (4/04)
  - 200 GFLOPS, 24 pipes (6/05)
  - 400 GFLOPS, 48 pipes (1/06)

- Small scale work in scientific apps
  - real-time Navier-Stokes simulations
  - particle simulations

- A few recent results vs CPU
  - 400x for static imagery geo-registration
  - 60x for video geo-registration
  - 100x for convolution
    - e.g. sharpen, blur, edge detection
  - 26x for hyperspectal covariance
  - 17x for discrete cosine transform
  - 4x for singular value decomposition

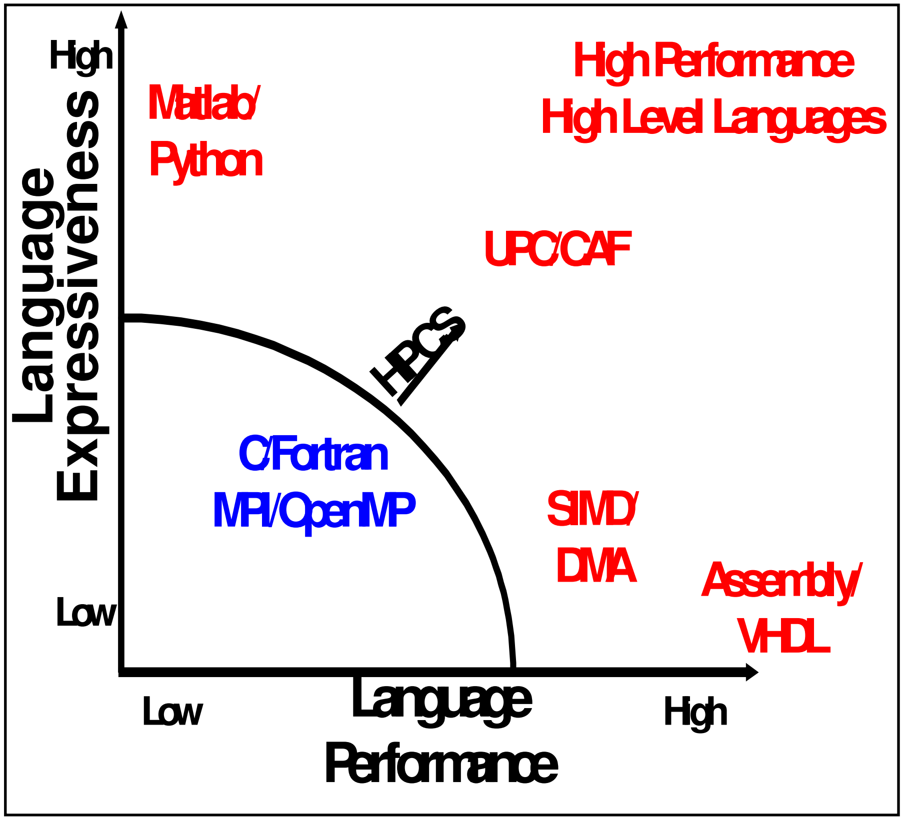
Recent and Ongoing Evaluations

- Cray X1
  - P.A. Agarwal, R.A. Alexander et al., “Cray X1 Evaluation Status Report,” ORNL, Oak Ridge, TN, Technical Report ORNL/TM-2004/13, 2004.
  - T.H. Dunigan, Jr., M.R. Fahey et al., “Early Evaluation of the Cray X1,” Proc. ACM/IEEE Conference High Performance Networking and Computing (SC03), 2003.
  - T.H. Dunigan, Jr., J.S. Vetter et al., “Performance Evaluation of the Cray X1 Distributed Shared Memory Architecture,” IEEE Micro, 25(1):30-40, 2005.
- Cray XD1
  - M.R. Fahey, S.R. Alam et al., “Early Evaluation of the Cray XD1,” Proc. Cray User Group Meeting, 2005, pp. 12.
- Cray XT3
  - J. S. Vetter, S. R. Alam et al., “Early Evaluation of the Cray XT3 at ORNL,” Proc. Cray User Group Meeting, 2005, pp. 12.
- SGI Altix
  - T.H. Dunigan, Jr., J.S. Vetter, and P.H. Worley, “Performance Evaluation of the SGI Altix 3700,” Proc. International Conf. Parallel Processing (ICPP), 2005.
- SRC
  - M.C. Smith, J.S. Vetter, and X. Liang, “Accelerating Scientific Applications with the SRC-6 Reconfigurable Computer: Methodologies and Analysis,” Proc. Reconfigurable Architectures Workshop (RAW), 2005.

- Underway
  - XD1 FPGAs
  - ClearSpeed
  - EnLight
  - Multicore processors: AMD, Intel
  - IBM BlueGene/L

Programming Challenges

- ⇨ Portability
  - – Unique software stack and programming model for each alternative
- ⇨ Programmer Productivity
- ⇨ Code Maintenance
- ⇨ Improving sustained performance is equivalent to improving the application to architecture mapping
- ⇨ Data Management
  - – Bandwidth, latency challenges
- ⇨ Cost
  - – These systems must add measurable value



Source: DARPA HPCS Productivity Team

Common Programming Models

- ⇨ Explicit message passing -- e.g. MPI
- ⇨ Explicit one sided communication -- e.g. SHMEM.
- ⇨ Compiler directives -- e.g. OpenMP
- ⇨ Explicit threading models -- e.g. pthreads, sproc

- ⇨ Here to stay… for now
- ⇨ But can we do better?
  - I.e. provide code developers with better ways (faster development, strong performance)

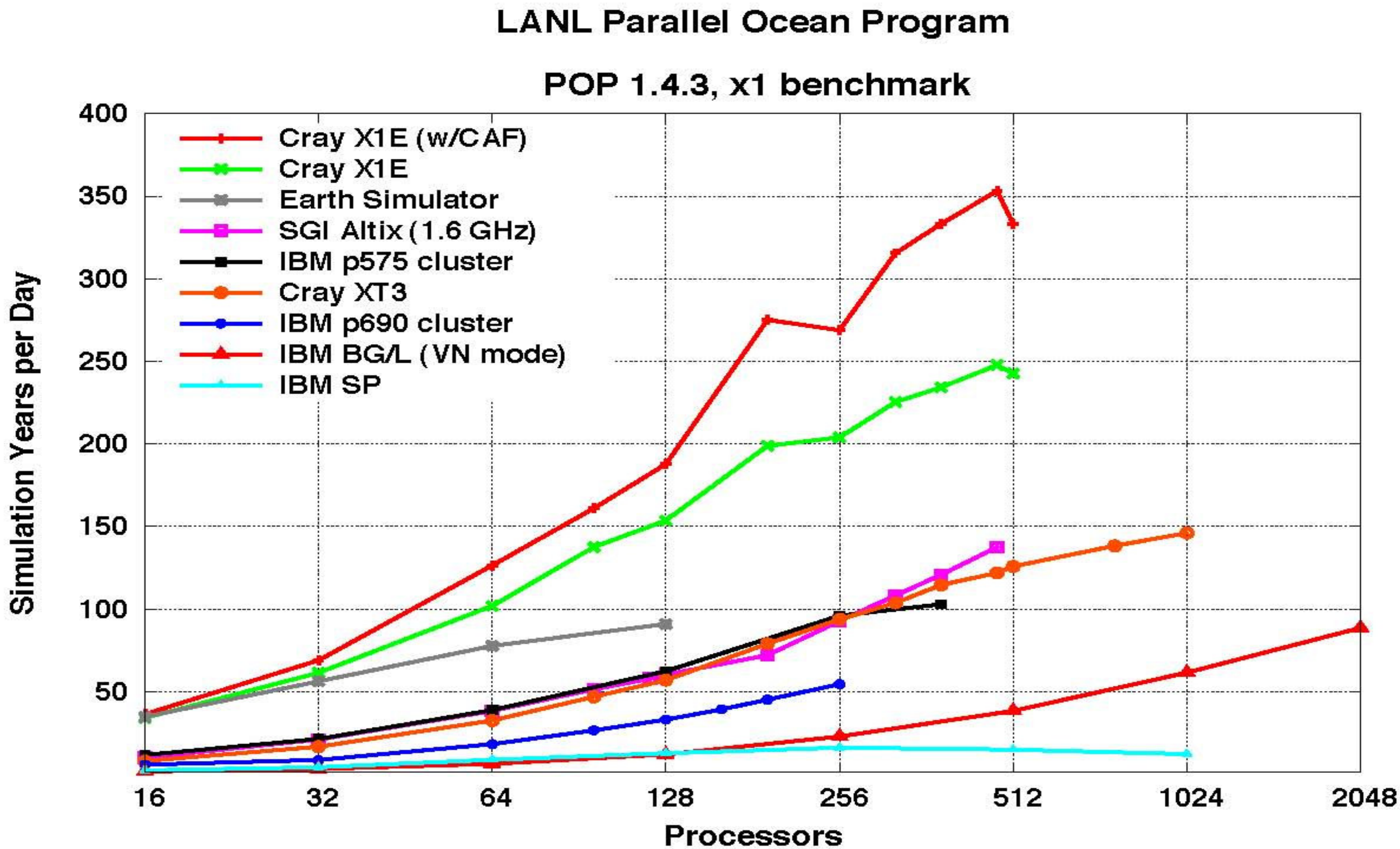
Emerging Programming Models

- ⇨ Partitioned Global Address Space languages.
  - Small set of extensions to existing languages enable parallelism designed to create a global address space, even on machines that don't physically have one.
- ⇨ CoArray Fortran (CAF):
  - SPMD
  - User must deal with data distribution.
  - Fortran2008 inclusion.
- ⇨ Unified Parallel C (UPC)
  - C like model, with data distribution (mostly) hidden from user.
  - Random memory access model (NSA driven)
- ⇨ Alternates: "Global Arrays" (PNNL), etc.

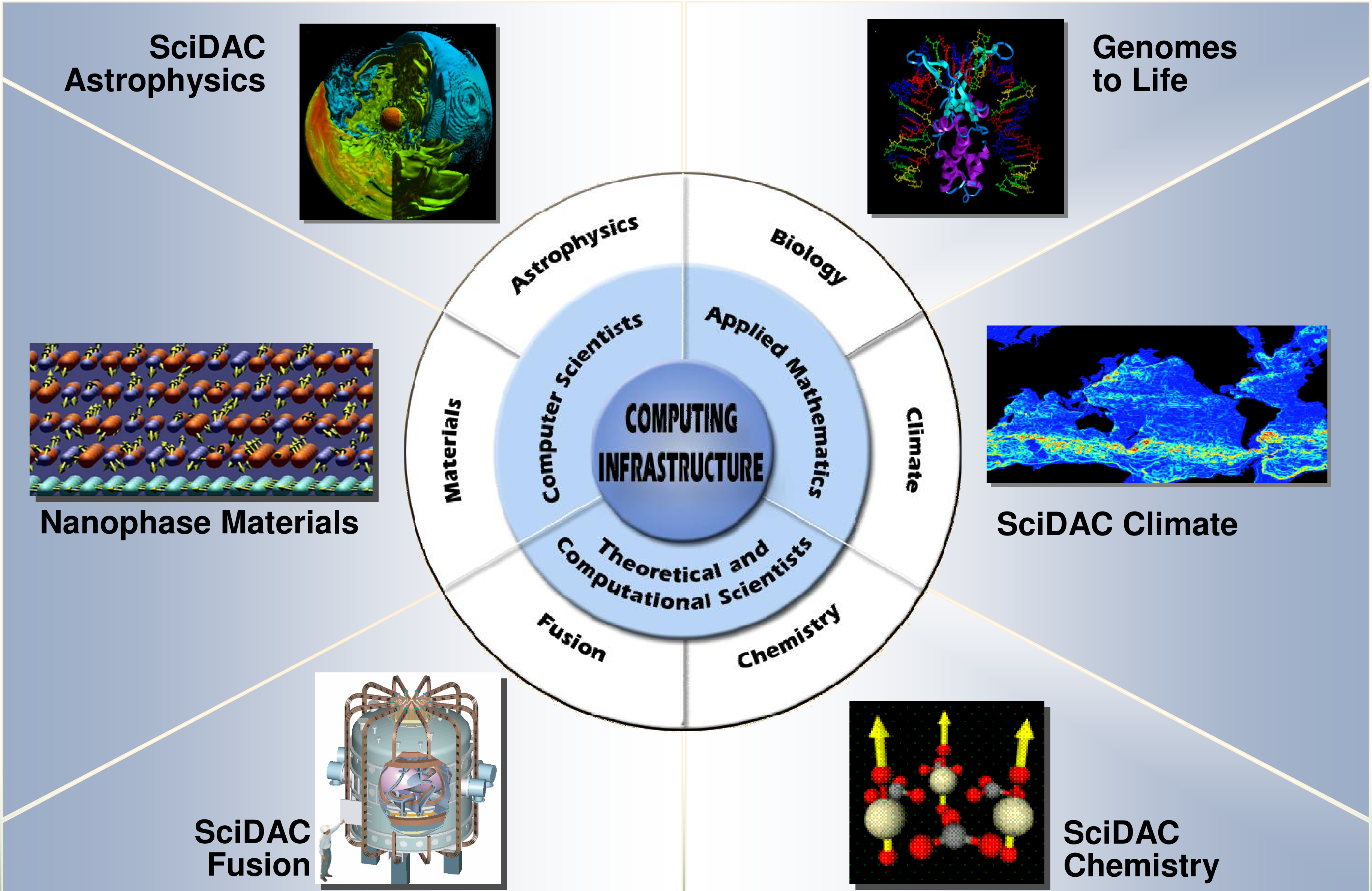
Next Generation Programming Models

- DARPA HPCS parallel processing languages (2010).
  - independent of architecture program.
- Chapel (Cray, with JPC/CalTech)
  - High level multithreaded model
  - supports data, task, and nested parallelism.
- X10 (IBM)
  - OO, Java-like.
- Fortress (Sun)
  - "Java for scientists on peta-scale architectures." -- Guy Steele (co-author of Java)
- Other players:
  - Titanium: Java-based, SPMD (UC Berkeley)

Application Impact – Fortran CoArrays



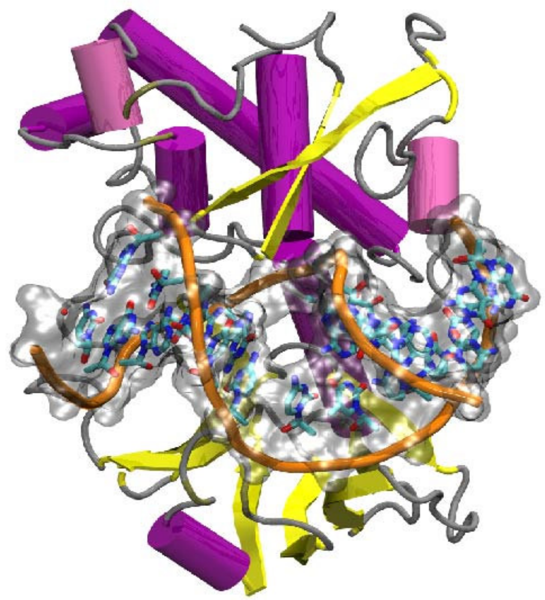
ORNL has Major Efforts Focusing on Grand Challenge Scientific Applications

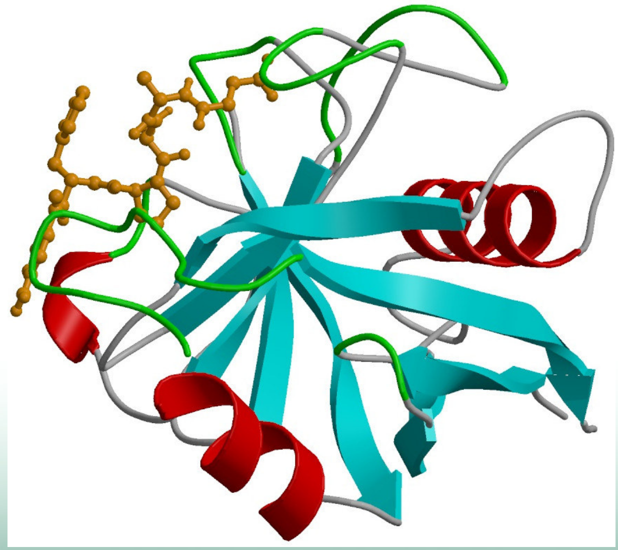


Case Study of a Life Sciences Application

Computational Biology using Molecular Modeling

- Wide community of biologist are interested in the multi-scale modeling of biomolecules
- Structure – Dynamics – Function
- Spans multiple scales of time and space
- *Multi-scale modeling of a real system may require **1 peta-flop/s** for an entire year!*
- Scaling of existing software packages and algorithms is limited





Joint work between Sadaf Alam and Comp Biologist Pratul Agarwal at ORNL.

Computer Simulations
(Molecular Dynamics)

⇨ Mathematical (potential) function

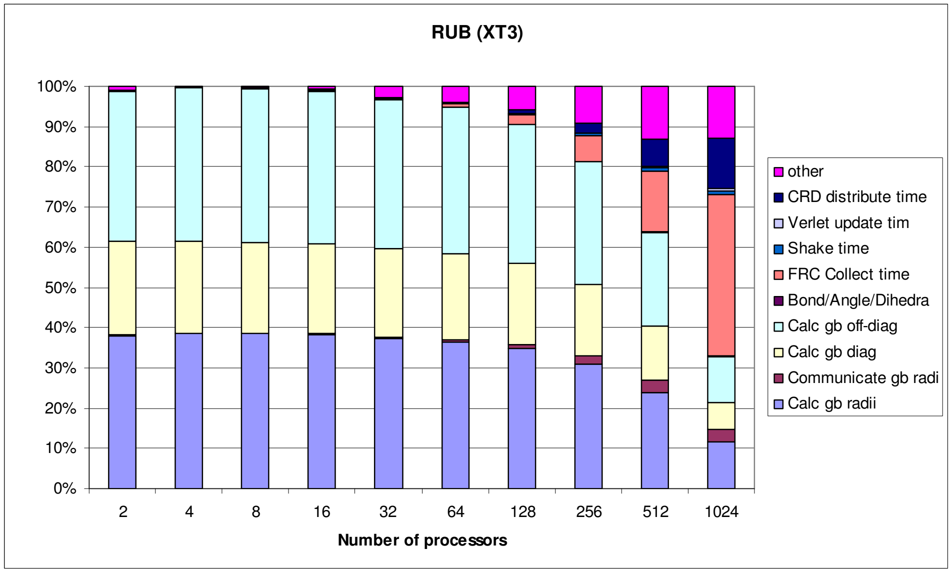
$$V(r^n) = \sum_{bonds} K_l (l - l_{eq})^2 + \sum_{angles} K_\theta (\theta - \theta_{eq})^2 + \sum_{torsions} \frac{V_n}{2}(1 + \cos[n\phi - \gamma]) + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(\left[\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right] + \frac{q_i q_j}{\varepsilon r_{ij}} \right)$$

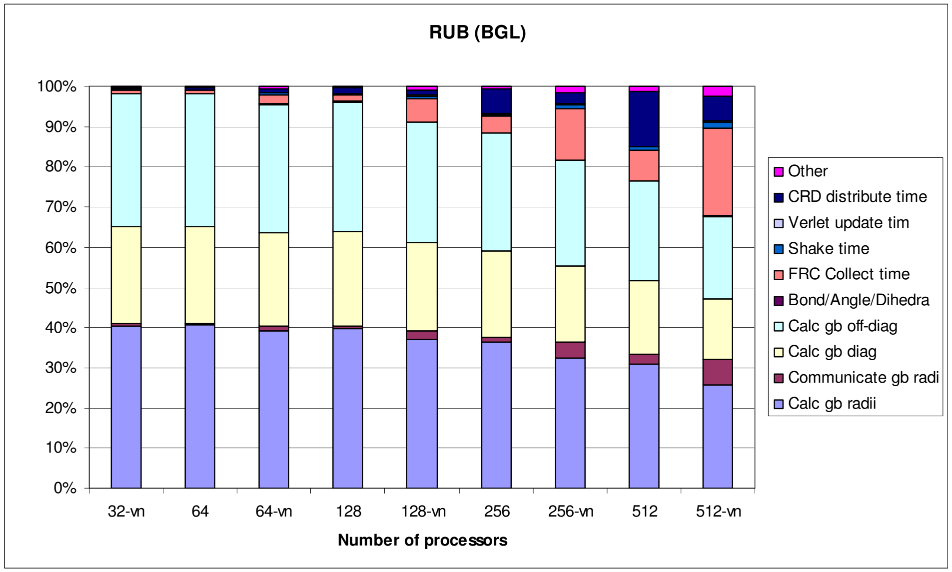
- Bond stretching, angle bending, angle torsion and the non-bond term
- Degree of freedom = 3N-6, where N=number of atoms
- Number of points to sample=M^{3N-6}, M >> 10
- Packages: Amber, GROMACS, GAMESS, LAMMPS, NAMD

AMBER Performance Analysis

- ORNL Computational biologists were using AMBER for their simulations, but its scalability was limited to about 128 processors
- Used several tools to study AMBER's performance
  - MPIP, PAPI, Xprofiler, GPROF
- Modified communication operations to improve scaling
- Identified computational kernels for acceleration with FPGAs

AMBER Profiling on Cray XT3 and IBM BlueGene/L





XT3

Bottlenecks: Distribute, Collect and I/O times

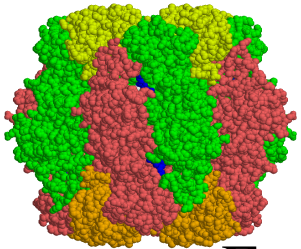
Expected to improve significantly as system matures

BGL

Bottlenecks: Distribute and collect times

Computation and communication times can improve with tool chain

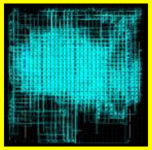
Amber Control Flow for RUB
(RuBisCO with Generalized Born method)

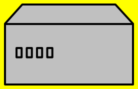


RunMD

Main method of Sander

- Force
  - Non-bond
    - Generalized Born
      - GB Computation
      - GB Communication
  - Bond
  - FRC Collect
- Shake
- Update
- CRD distribute



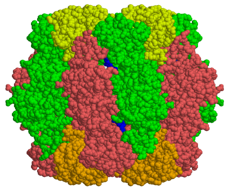


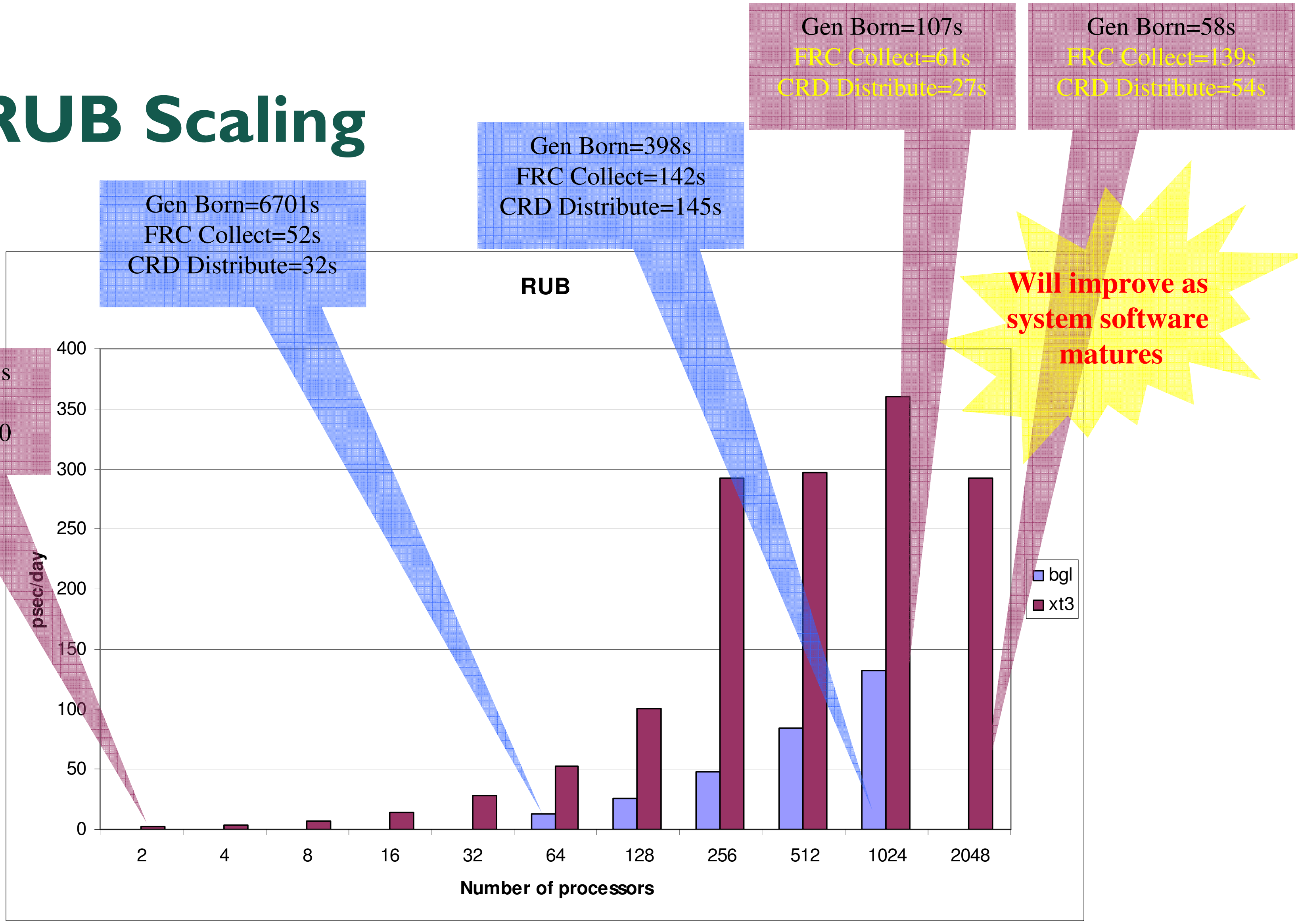
Short_ene: mapped to FPGA
Divided and distributed in MPI model

Cost increases with number of processors

Most expensive on a single node system

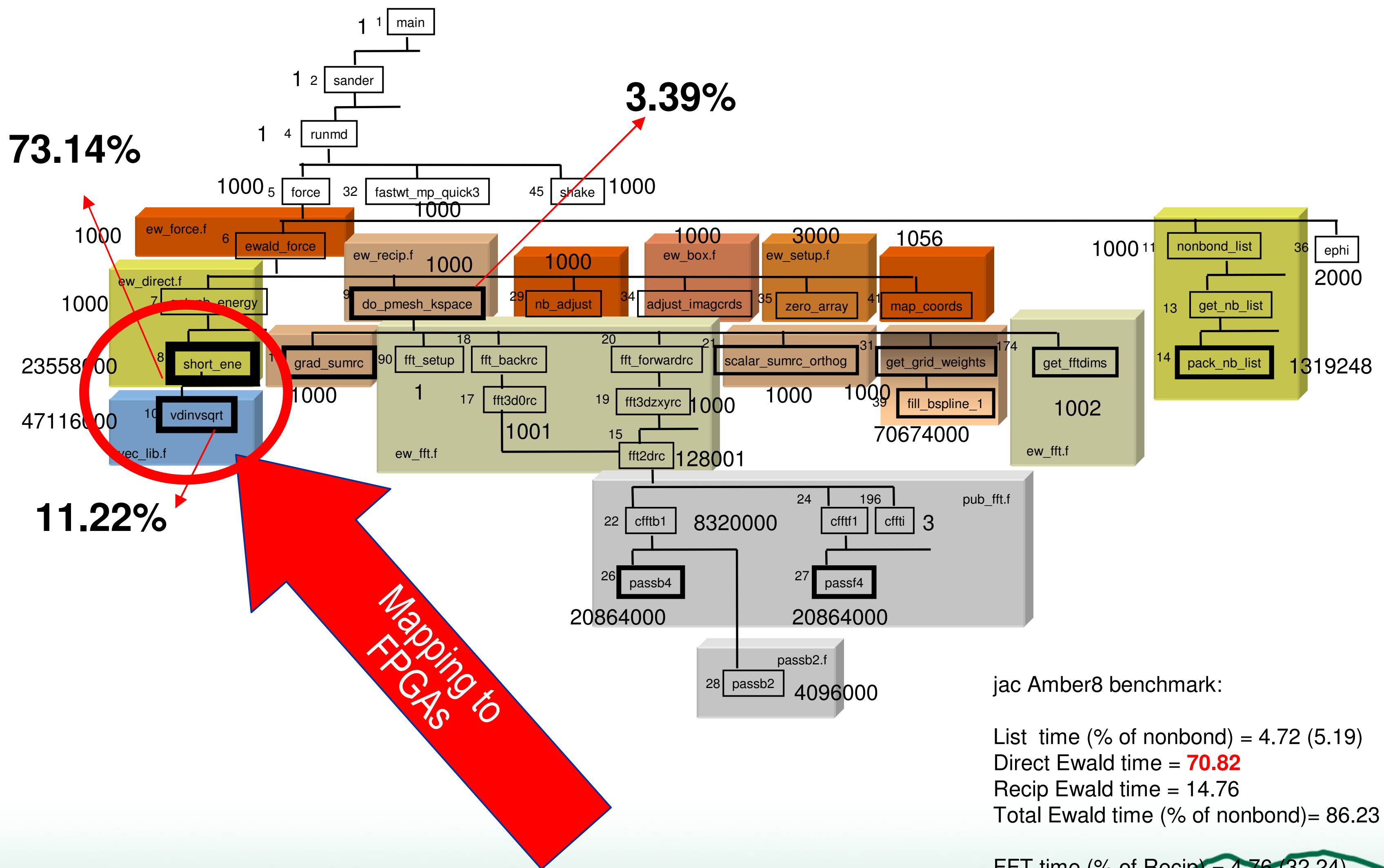
RUB Scaling





Rubisco with Generalized Born solvation method (ORNLtest3). Note that on BGL only results from 64, 128, 256, 512 nodes run are shown.

Mapping Amber Kernel to FPGAs



jac Amber8 benchmark:

List time (% of nonbond) = 4.72 (5.19)
Direct Ewald time = **70.82**
Recip Ewald time = 14.76
Total Ewald time (% of nonbond)= 86.23 (**94.81**)

FFT time (% of Recip) = 4.76 (32.24)

AMBER Summary

- ⇨ Performance analysis identified communication components as limiting scalability
- ⇨ Improved by code modifications
- ⇨ Amber scaling was limited to 128 nodes but now we have run experiments on 1024 nodes on Bluegene/L and on 2048 nodes on Cray XT3
- ⇨ Achieved close to order of a nano-second/day on early evaluation stage supercomputing systems
- ⇨ Mapping compute intensive kernel to SRC MapStation (a reconfigurable computing system)

Summary

- ⇨ We are analyzing HPCS applications to help understand current and future requirements
- ⇨ Generating empirical Sequoia traces from large scale experiments
  - Developed trace analysis tools to help understand communication scaling
- ⇨ Developing toolkit that we can distribute that will allow
  - Creation of symbolic models that can be evaluated in traditional environments like MATLAB or Python
  - Projections to larger scale
  - Sensitivity analysis
  - Allows users to model and validate their applications
  - Adding capabilities to allow time transformation

Acknowledgements and More Info

- This research was sponsored by the Office of Mathematical, Information, and Computational Sciences, Office of Science, U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Batelle, LLC. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

- http://www.csm.ornl.gov/ft
- http://www.csm.ornl.gov/evaluation
- Email: vetter@ornl.gov